(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 December 2000 (28.12.2000)

PCT

(10) International Publication Number
WO 00/79831 A1

(51) International Patent Classification⁷: H04Q 11/04, H04L 1/00, 29/06

(21) International Application Number: PCT/SE00/01335

(22) International Filing Date: 21 June 2000 (21.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
9902403-6 23 June 1999 (23.06.1999) SE

(71) Applicant *(for all designated States except US)*: TERACOM AB [SE/SE]; Box 17666, S-118 92 Stockholm (SE).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: HAGGÅRD LJUNGQUIST, Jonas [SE/SE]; Spångavägen 70, S-168 75 Bromma (SE).

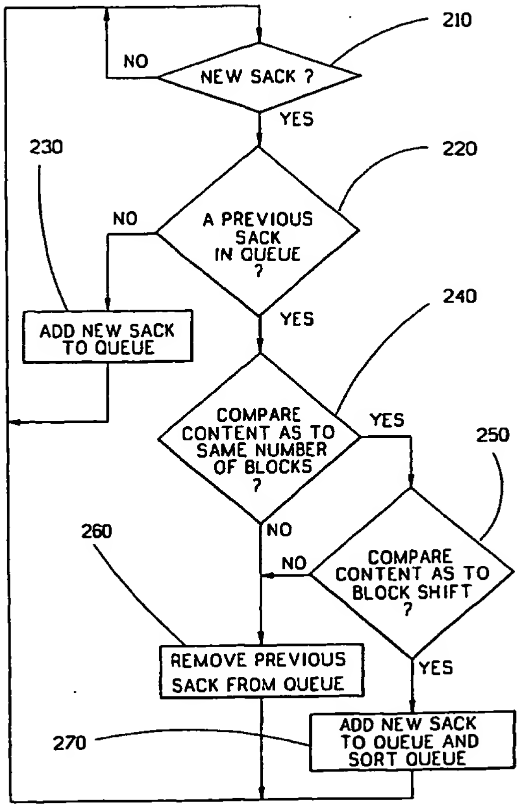
(74) Agents: BERGENTALL, Annika et al.; Cegumark AB, Box 53047, S-400 14 Göteborg (SE).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,

[Continued on next page]

(54) Title: METHOD FOR FLOW CONTROL


210
NO
NEW SACK ?
YES
230
220
NO
A PREVIOUS SACK IN QUEUE ?
ADD NEW SACK TO QUEUE
YES
240
COMPARE CONTENT AS TO SAME NUMBER OF BLOCKS ?
YES
250
260
NO
NO
COMPARE CONTENT AS TO BLOCK SHIFT ?
REMOVE PREVIOUS SACK FROM QUEUE
YES
270
ADD NEW SACK TO QUEUE AND SORT QUEUE

(57) Abstract: A method of efficiently using Transmission Control Protocol (TCP) with Acknowledgements (ACKs) comprising Selective Acknowledgement (SACK) options in asymmetrical networks by reducing the number of ACKs with SACK options to be transferred back to a sender. New ACKs with SACK options are compared to previous ACKs with SACK options as to their content, and in dependence of the comparison, i.e. if the previous ACK with a SACK option comprises redundant or invalid information and no additional information in relation to the new ACK with a SACK option, then the previous ACK with the SACK option is removed and not sent back to a sender.

IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *With international search report.*

— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

Method for flow control

TECHNICAL FIELD

The present invention generally relates to a method of handling asymmetrical transmissions, specifically information transported by means of transmission control protocol (TCP) via asymmetrical error prone channels.

BACKGROUND

A demand for mobile high capacity data transfers have put high capacity, but unfortunately asymmetrical, wireless broadcasting transmission channels in focus. The majority of all information transfers today are made with the help of Transmission Control Protocol (TCP). Unfortunately this protocol is not designed for neither asymmetrical transmission or wireless error prone transmissions, such as Digital Audio Broadcasting (DAB), Digital Video Broadcasting Terrestial (DVB-T), and Wavelan. Due to this there has come about many different suggestions such as replacing TCP with other modern protocols and modifying TCP to different varying degrees. For information/data transfer via wireless transfer channels special measures have to be taken to be able to manage a low as possible transmitting power. The measures can, for example, be the use of error correcting codes or special retransmission protocols.

To be able to handle bit errors with TCP, a Selective Acknowledgement (SACK) option to Acknowledgements (ACKs) have been developed. By means of the SACK option the TCP transmitter can get to know which packets/segments after the first lost segment that have reached their destination. A SACK option can normally give information of up to three contiguous blocks of segments that are correctly received and also three contiguous blocks of missing or erroneous segments in the information transmission. The use of SACK options presupposes that there is no significant asymmetry in the network so that as many SACKs as information segments transmitted downlink, can be transmitted uplink.

One way of dealing with asymmetry in an extremely asymmetrical network is called ACK-filtering and works on the principles that many of the ACKs that are transmitted back to the transmitter are redundant and do not have to be transmitted. If there is a limitation in the return channel there will emerge a queue and it is in this queue that the ACK-filtering operates.

A combination of these two methods will create problems. SACK options are based on the assumption that there is no limitation in the return channel, and they are thus designed accordingly, and if an ACK-filtering is operated on a SACK queue, i.e. a queue of ACKs with SACK options, then some SACK blocks will be completely removed and then the transmitter will not find out that these blocks have been received. This will lead to the transmitter erroneously retransmitting the segments in these blocks that have already been received. This will result in the downlink transmission channel suffering when bandwidth is used for unnecessary retransmissions.

The problems associated with asymmetrical networks in conjunction with TCP needs to be solved satisfactorily. TCP can handle an asymmetry factor of 50 times the downlink to the uplink in most common implementations. Future networks can have an asymmetry factor of around a 1000 times. There seems to be room for improvement.

SUMMARY

An object of the invention is to define a method for efficiently transferring information in an asymmetrical network.

Another object of the invention is to define a method of efficiently using TCP with ACKs comprising SACK options in an asymmetrical network.

The aforementioned objects are achieved according to the invention by a method of reducing the number of ACKs with SACK options to be transferred back to a sender. New ACKs with SACK options are compared to previous ACKs with SACK options as to their content, and in dependence of the comparison, i.e. if the previous ACK with a SACK option comprises redundant or invalid information and no additional information in relation to the new ACK with a SACK option, then the previous ACK with the SACK option is removed and not sent back to a sender.

The aforementioned objects are also achieved by a method of, when receiving information in a network that transports information according to the general principles of the Transmission Control Protocol (TCP), reducing the number of selective acknowledgements (SACKs) to be transferred back to a sender. Selective Acknowledgements, SACKs, will generally be used for referring to Acknowledgements (ACKs) with Selective Acknowledgement options. The method comprises a plurality of steps when a new selective acknowledgement has been triggered/generated. In a first step determining if there is a previous selective acknowledgement in a queue to be transferred to the sender, and if there is a previous selective acknowledgemnt in the queue then further performing the following additional steps. In a first additional step comparing the content of the new selective acknowledgement with the content of the previous selective acknowledgement. Finally in a second additional step removing the previous selective acknowledgement from the queue in dependence on the result of the comparison.

Advantageously in the second additional step of removing the previous selective acknowledgement from the queue, the previous selective acknowledgement is removed from the queue if the previous selective acknowledgement has a different number of blocks than the new selective acknowledgement. Further in the second additional step of removing the previous selective acknowledgement from the queue, the previous selective acknowledgement can and/or also advantageously be removed from the queue if the previous selective acknowledgement has the same number of blocks as the new selective acknowledgement and if only the right edge of the first block is different between the previous selective acknowledgement and the new selective acknowledgement.

In some versions of the invention the method can further advantageously comprise the additional step of sorting the selective acknowledgements in the queue as to which blocks are redundant. Preferably the step of sorting comprises the following four block sorting steps. A first block sorting step of disassembling into blocks the selective acknowledgements that have not been transferred to a sender. A second block sorting step of removing redundant blocks. A third block sorting step of assembling the remaining blocks into block sorted selective acknowledgements. A fourth block sorting step of placing the block sorted selective acknowledgements into the queue. The new selective acknowledgement is preferably not disassembled, but the knowledge of which blocks are comprised in the new selective acknowledgement is used in the second block sorting step for removing redundant blocks. In some versions there are status indicators to keep track of one or both of if and when a queue has been sorted with regard to selective acknowledgements.

The step of sorting can in some version of the method alternatively comprise the following two selective acknowledgement steps. A first selective acknowledgement step of comparing the selective acknowledgements that have not been transferred to a sender. A second selective acknowledgement step of removing the redundant selective acknowledgements, i.e. removing selective acknowledgements only comprising redundant blocks that are included in other selective acknowledgements.

In some versions of the method the step of sorting is performed only if the previous selective acknowledgement has not been removed from the queue.

Preferably the method is performed on the output queue in any suitable layer from the TCP layer to the link layer, specifically the method is performed most advantageously on the internet protocol layer output queue.

By providing a method for removing Selective Acknowledgements (SACKs) from an output queue a plurality of advantages over prior art systems are obtained. A primary purpose of the invention is to, in a simple manner, enable the use of highly asymmetrical networks. According to the invention this is achieved primarily by making simple, and thus quickly executable, comparisons between the contents of new SACKs and previous SACKs and removing a previous SACK if the content is redundant and/or invalid. There is thus no loss of SACK information which could lead to unnecessary retransmissions. By reducing the number of SACKs that have to be sent back, a network with a higher asymmetry factor can be used. Other advantages of this invention will become apparent from the description.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will now be described in more detail for explanatory, and in no sense limiting, purposes, with reference to the following figures, in which

Fig. 1 shows an asymmetrical transmission system,

Fig. 2 shows a flow chart of a method according to the invention,

Fig. 3 shows the layered architecture of an information consumer according to the invention.

DETAILED DESCRIPTION

In order to clarify the method and device according to the invention, some examples of its use will now be described in connection with Figures 1 to 3.

The invention concerns information transfers in networks, especially asymmetrical networks. Figure 1 shows an asymmetrical transmission network on which the invention is intended to operate. An information consumer 150 communicates with a first uplink gate 160 which transfers download data requests via a low capacity back channel 170 to a second uplink gate 180 which transfers requests to a connection network 120 such as internet, onto which a desired information provider 110 is connected. The low capacity back/uplink channel 170 might be a telephone network or a mobile cellular mobile telephone network in which case the first uplink gate 160 and the second uplink gate 180 can be modems. After having received the request and information on how to transfer the requested information to the information consumer 150, the information provider 110 will transfer the desired information, if it is available, to the information consumer 150 via the connection network 120, a downlink gate 130 and a high capacity downlink channel 140. The high capacity downlink channel can, for example, be a high capacity broadcasting system such as a Digital Audio Broadcasting (DAB) system, a Digital Video Broadcasting Terrestial (DVB-T) system, or a cable TV network.

As mentioned previously, it is advantageous to use ACKs with SACK options, which will simply be referred to as SACKs. With SACKs an information consumer 150 can inform the sender about all segments that have arrived successfully, so the sender need retransmit only the segments that have actually been lost. A SACK can normally comprise information of up to three contiguous blocks of segments that are correctly received and also up to three contiguous blocks of missing or erroneous segments in the information transmission. The first block of a SACK always relate to the segment that triggered it. Each block represents received bytes of data that are contiguous and isolated, i.e. the bytes below the block (Left Edge of Block minus 1), and just above the block (Right Edge of

Block), have not been received. The Left Edge of a Block is the first sequence number of this block and the Right Edge of the Block is the sequence number immediately following the last sequence number of this block.

Following is a short example of how an ACK with a SACK option works. It is assumed that the left window edge is 5000 and that the data transmitter sends a burst of 8 segments, each containing 500 data bytes.

Sent	Received/ Trigger	ACK	SACK	option	blocks			
			First Left Edge	First Right Edge	Second Left Edge	Second Right Edge	Third Left Edge	Third Right Edge
5000	5000	5500						
5500								
6000	6000	5500	6000	6500				
6500								
7000	7000	5500	7000	7500	6000	6500		
7500								
8000	8000	5500	8000	8500	7000	7500	6000	6500

The first segment, 5000, is received and triggers an ACK 5500 indicating that segment 5500 is not received. The second segment, 5500, is lost, nothing further happens. The third segment, 6000, is recieved and triggers an ACK 5500 [6000 6500 ; -- ; --], i.e. an ACK with a SACK option. This indicates to the transmitter that the second segment is lost and by means of the SACK option that segment 6000 (bytes 6000 to 6499) has been received and not segment 6500 and further. The fourth segment, 6500, is lost, no reaction. The fifth segment, 7000, is received and triggers an ACK 5500 [7000 7500 ; 6000 6500 ; --]. The sixth segment, 7500, is lost, no action. The seventh segment, 8000, is received and triggers an ACK 5500 [8000 8500 ; 7000 7500 ; 6000 6500].

SACKs presupposes that the network used for information transfer is fairly symmetrical, i.e. that there is no practical limitations in the uplink channel 170 for transferring SACKs back to the sender. In an asymmetrical network according to Fig. 1 an output queue 155 full of SACKs will form at or around the information consumer 150 or the first uplink gate 160 due to limitations in the uplink channel 170. This will cause delays in the transfer of SACKs to the downlink gate 130 and/or the information provider 110. This in turn will also put restrictions on the transfer of information on the downlink channel 140, because there are restrictions on how much information is sent without any confirmation of it being received or not. There is thus a need to eliminate any unnecessary delays returning SACKs. According to the invention, SACKs are compared as to their content to determine if they can be discarded or not. This is preferably performed on the SACKs in the output queue 155.

When the SACKs are compared as to their content three different cases arise. A first case is when the new SACK has more blocks than the previous SACK. A second case is when the new SACK has fewer blocks than the previous SACK. Finally a third case is when the new SACK has the same number of blocks as the previous SACK.

The first case when the new SACK has more blocks than the previous SACK, then the previous SACK is redundant and can be removed. SACKs use a stack principle with push and pop where it always tries to report the latest blocks. This means that the new SACK will comprise all of the blocks of the previous SACK and one or more new blocks. The previous SACK is thus redundant in that it does not contain any information that is not in the new SACK. For example a previous SACK has been triggered, ACK 10 [18 19 ; 11 17 ; --], segments 20 to 24 arrives triggering a new SACK, ACK 10 [20 25 ; 18 19 ; 11 17], making the previous SACK redundant.

The second case when the new SACK has fewer blocks than the previous SACK, then the previous SACK contains old invalid information and can be removed. Because SACKs always comprise the maximum number of blocks that is possible (in most implementations it is three blocks) this means that if the new SACK comprises fewer blocks, then new data has arrived to the receiver and filled one of the holes of data. The previous SACK thus comprises old invalid information and can be removed. For example a previous SACK is ACK 10 [20 25 ; 18 19 ; 11 17], the receiver then receives segment 19 triggering a new SACK, ACK 10 [18 25 ; 11 17 ; --], making the previous SACK redundant.

Finally a third case is when the new SACK has the same number of blocks as the previous SACK. This case needs to be divided into two different events. A first event is when new data is added to the first block. The only thing that is different between the two SACKs is that the first block is different. This means that the previous SACK comprises old invalid information and can be removed. For example a previous SACK is ACK 10 [20 25 ; 18 19 ; 11 17], the receiver receives segment 25 triggering a new SACK, ACK 10 [20 26 ; 18 19 ; 11 17], making the previous SACK redundant. A second event is when new data give rise to a new block. The new block will be put first in the SACK and push out the last block, i.e. the last block will be shifted out and the previous SACK can thus not be removed. For example a previous SACK is ACK 10 [20 25 ; 18 19 ; 11 17], the receiver receives segment 28 triggering a new SACK, ACK 10 [28 29 ; 20 25 ; 18 19], i.e. the previous SACK comprises information about segments 11 to 17 that is not comprised in the new SACK, the previous SACK can thus not be removed.

Additionally the SACKs in an output queue can be sorted as to redundant SACKs and blocks, this is especially advantageous in the cases when a previous SACK cannot be removed. According to the invention this can be done in two different ways. In a first output queue redundancy sorting, all the blocks of the SACKs in the queue are put in a separate place/queue where all the redundant blocks are removed, whereafter new SACKs are put together with the remaining non-redundant blocks and put into the output queue. A newly triggered SACK is preferably not stripped of its blocks but only used for the determination of which blocks are redundant. In a second, and preferred, output queue redundancy sorting, the blocks of the SACKs in the output queue are compared to thus be able to remove SACKs that are completely redundant.

Figure 2 shows a flow chart of a method according to the invention. A first step 210 awaits the generation/triggering of a new SACK. When a new SACK has been generated to be transmitted back to a sender then the procedure determines in a second step 220 if there is any previous SACKs in an output queue which comprises SACKs to be transmitted back to the sender. If there are no SACKs in the output queue then the procedure continues with a third step 230 that simply adds the new SACK to the output queue, whereafter the procedure returns to the first step 210. In the second step 220 if it is instead determined that there is a previous SACK in the output queue, then the procedure continues with a fourth step 240. The fourth step 240 compares the contents of the new SACK with the contents of the previous SACK, more specifically the number of blocks in the two SACKs are compared. If the two SACKs have the same number of blocks then the procedure continues with a fifth step 250, and if they have a different number of blocks then the procedure continues with a sixth step 260. The fifth step 250 also compares the contents of the new SACK with the contents of the previous SACK, more specifically in this step the contents of the two SACKs are compared to determine if there has been a block shift. If there has been a block shift then the procedure continues with a seventh step 270, otherwise, if there has not been a block shift, then the procedure continues with the sixth step 260. The sixth step 260 removes the previous SACK from the output queue, adds the new SACK to the output queue and then the procedure continues with the first step 210. The seventh step 270 adds the new SACK to the output queue and then attempts to remove redundant blocks from the SACKs in the output queue by sorting and removing either just redundant blocks or by removing redundant SACKs. Thereafter the procedure continues with the first step 210.

Figure 3 shows the layered data communication architecture of an information consumer according to the invention. A layer refers to a process or a device inside a computer system, which is designed to perform a specific function. An application 310 is for our purposes the information consumer that requests downloads of data information, for example. These requests are forwarded to the Transmission Control Protocol (TCP) layer 320, then to the Internet Protocol (IP) 330, down via interfaces such as modems with Point-to-Point Protocols (PPP) 340 or Ethernet 350 to output lines 360. The requested information will, hopefully, after a while appear on the input lines 370 and transported through all the layers up to the application layer 310. SACKs will then be sent back to the sender. The invention will work on any suitable output queue from the TCP layer 320 to after the link layer. Preferably the invention will sort away SACKs on the output queue 332 of the IP layer 330. It is normally not favourable to have the invention filter the SACKs after the data communication link layer since the the packets need to be unpacked which can be an extremely resource consuming measure due to for example the compression in the link layer 340, 350. The invention will then also be customized to one type of link layer 340, 350, i.e. if a switch is made from ethernet 350 to a PPP with a modem 340 then the system according to the invention has to be reconfigured.

The present invention can be put into apparatus-form either as pure hardware, as pure software or as a combination of hardware and software. If the method according to the invention is realised in the form of software, it can be completely independent or it can be one part of a larger program. The software can suitably be located in a general purpose computer, in a dedicated computer, or a distributed combination thereof.

As a summary, the invention can basically be described as a method of improving the throughput in asymmetrical networks by removing/filtering in an efficient manner completely redundant SACKs only.

The invention is not limited to the embodiments described above but may be varied within the scope of the appended patent claims.

FIG 1

110	information source/provider
120	arbitrary network such as internet
130	gateway
140	high speed downlink
150	information consumer
155	output queue
160	gateway
170	slow uplink
180	gateway

FIG 2

210	determine if there is a new SACK
220	determine if there is a previous SACK in the output queue
230	add new SACK to output queue
240	compare content as to the same number of blocks
250	compare content as to if there is a block shift
260	remove previous SACK from output queue
270	add new SACK to output queue and sort SACKs in output queue

FIG 3

310	application layer
320	TCP layer
330	IP layer
332	IP layer output queue with SACKs
340	PPP
350	Ethernet
360	output lines
370	input lines

CLAIMS

1. A method of, when receiving information in a network that transports information according to the general principles of the Transmission Control Protocol (TCP), reducing the number of selective acknowledgements (SACK) to be transferred to a sender, **characterized in that** the method comprises the following steps when a new selective acknowledgement has been generated:

- determining if there is a previous selective acknowledgement in a queue to be transferred to the sender, and if there is a previous selective acknowledgemnt in the queue then further performing the following additional steps;
  - comparing the content of the new selective acknowledgement with the content of the previous selective acknowledgement;
  - removing the previous selective acknowledgement from the queue in dependence on the result of the comparison.

2. The method according to claim 1, **characterized in that** in the additional step of removing the previous selective acknowledgement from the queue, the previous selective acknowledgement is removed from the queue if the previous selective acknowledgement has a different number of blocks than the new selective acknowledgement.

3. The method according to claim 1 or 2, **characterized in that** in the additional step of removing the previous selective acknowledgement from the queue, the previous

selective acknowledgement is removed from the queue if the previous selective acknowledgement has the same number of blocks as the new selective acknowledgement and if only the right edge of the first block is different between the previous selective acknowledgement and the new selective acknowledgement.

4. The method according to any one of claims 1 to 3, **characterized in that** the method further comprises the additional step of:

- sorting the selective acknowledgements in the queue as to which blocks are redundant.

5. The method according to claim 4, **characterized in that** the step of sorting comprises the following block sorting steps:

- disassembling into blocks the selective acknowledgements that have not been transferred to a sender;
- removing redundant blocks;
- assembling the remaining blocks into block sorted selective acknowledgements;
- placing the block sorted selective acknowledgements into the queue.

6. The method according to claim 4, **characterized in that** the step of sorting comprises the following selective acknowledgement steps:

- comparing the selective acknowledgements that have not been transferred to a sender;
- removing the redundant selective acknowledgements.

7. The method according to any one of claims 4 to 6, **characterized in that** the step of sorting is performed only if the previous selective acknowledgement has not been

removed from the queue.

8. The method according to any one of claims 1 to 7, **characterized in that** the method is performed on a queue in any suitable layer from the TCP layer to after the link layer.

9. The method according to claim 8, **characterized in that** the method is performed on the internet protocol layer output queue.

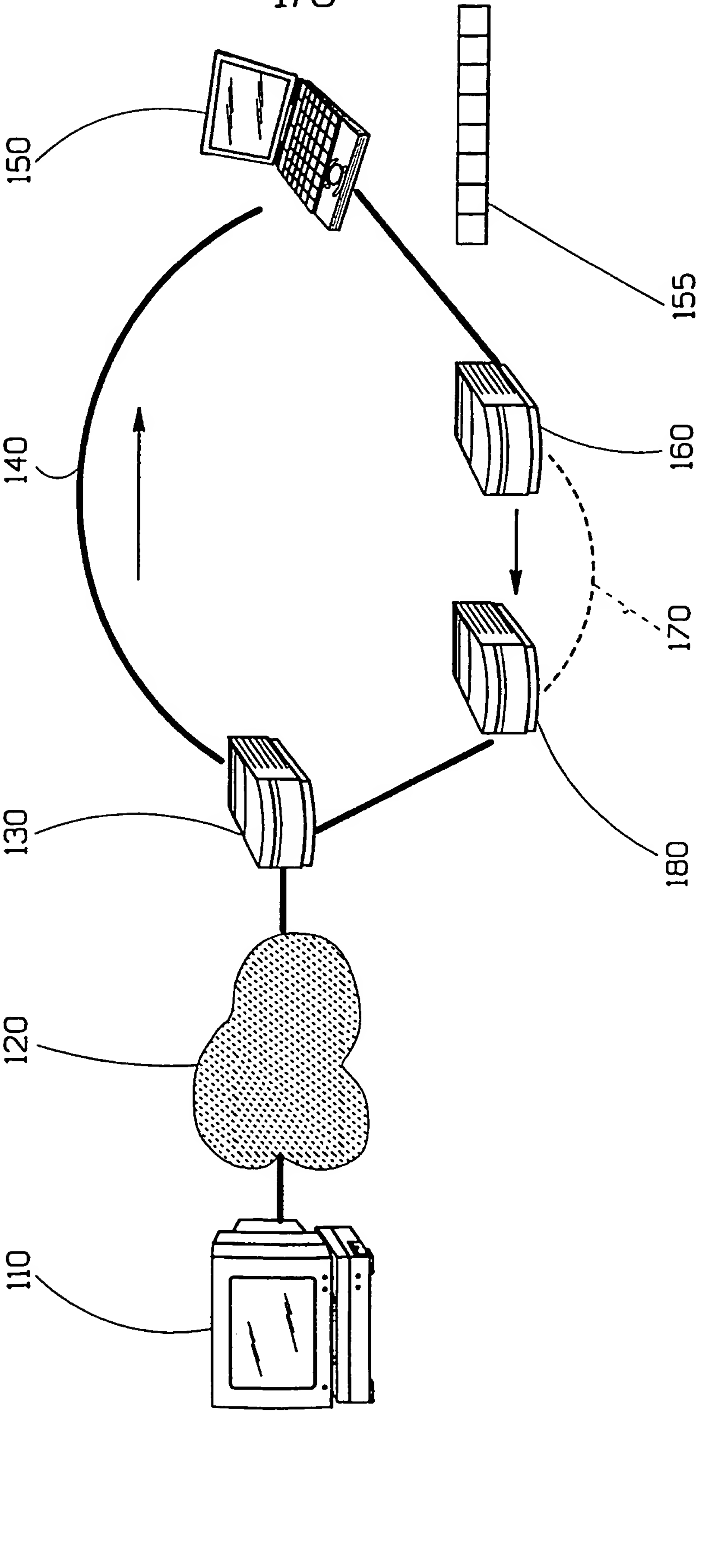


FIG.1

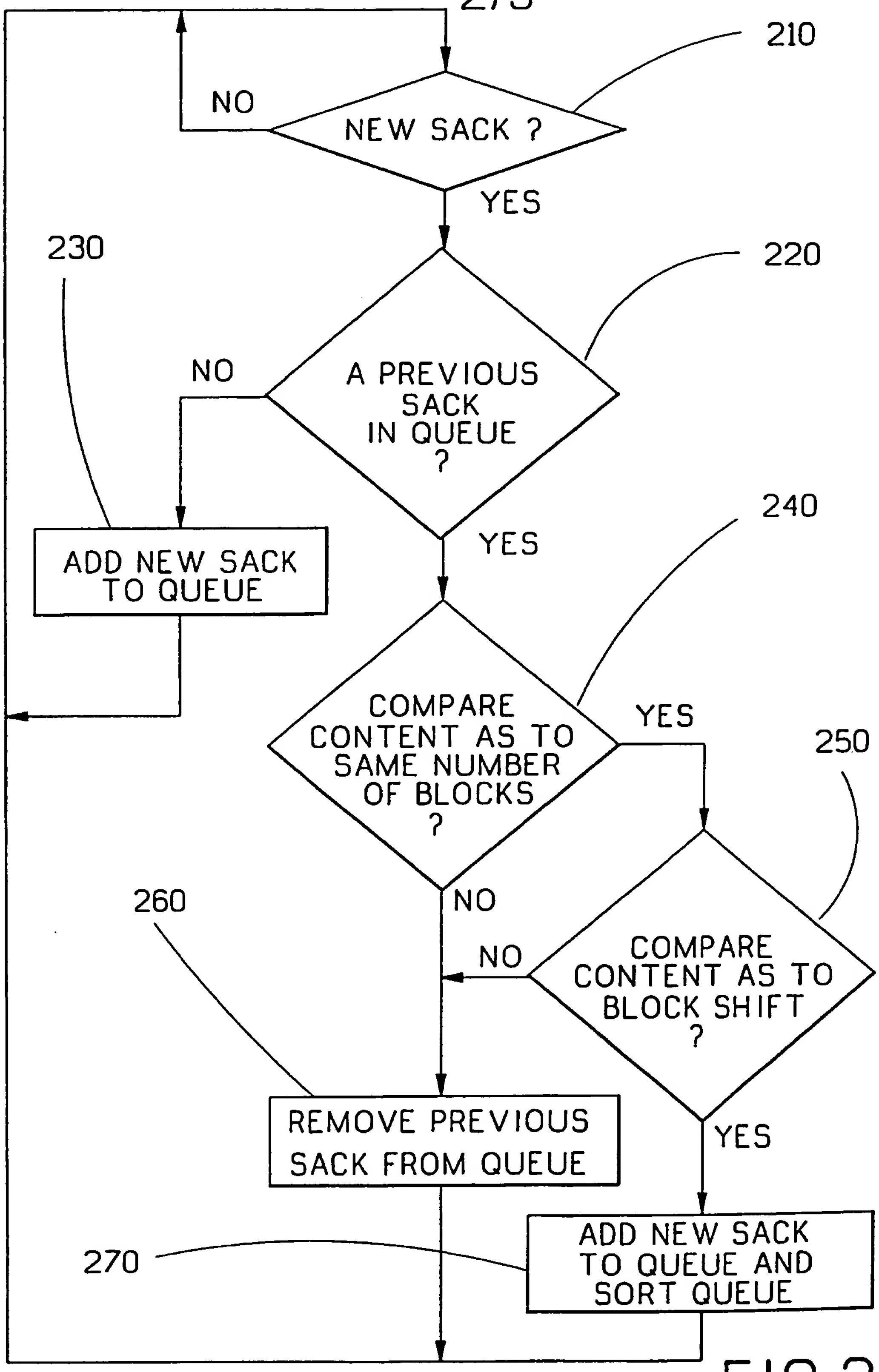

210
NO
NEW SACK ?
YES
230
220
NO
A PREVIOUS SACK IN QUEUE ?
YES
ADD NEW SACK TO QUEUE
240
COMPARE CONTENT AS TO SAME NUMBER OF BLOCKS ?
YES
250
260
NO
NO
COMPARE CONTENT AS TO BLOCK SHIFT ?
REMOVE PREVIOUS SACK FROM QUEUE
YES
270
ADD NEW SACK TO QUEUE AND SORT QUEUE

FIG.2

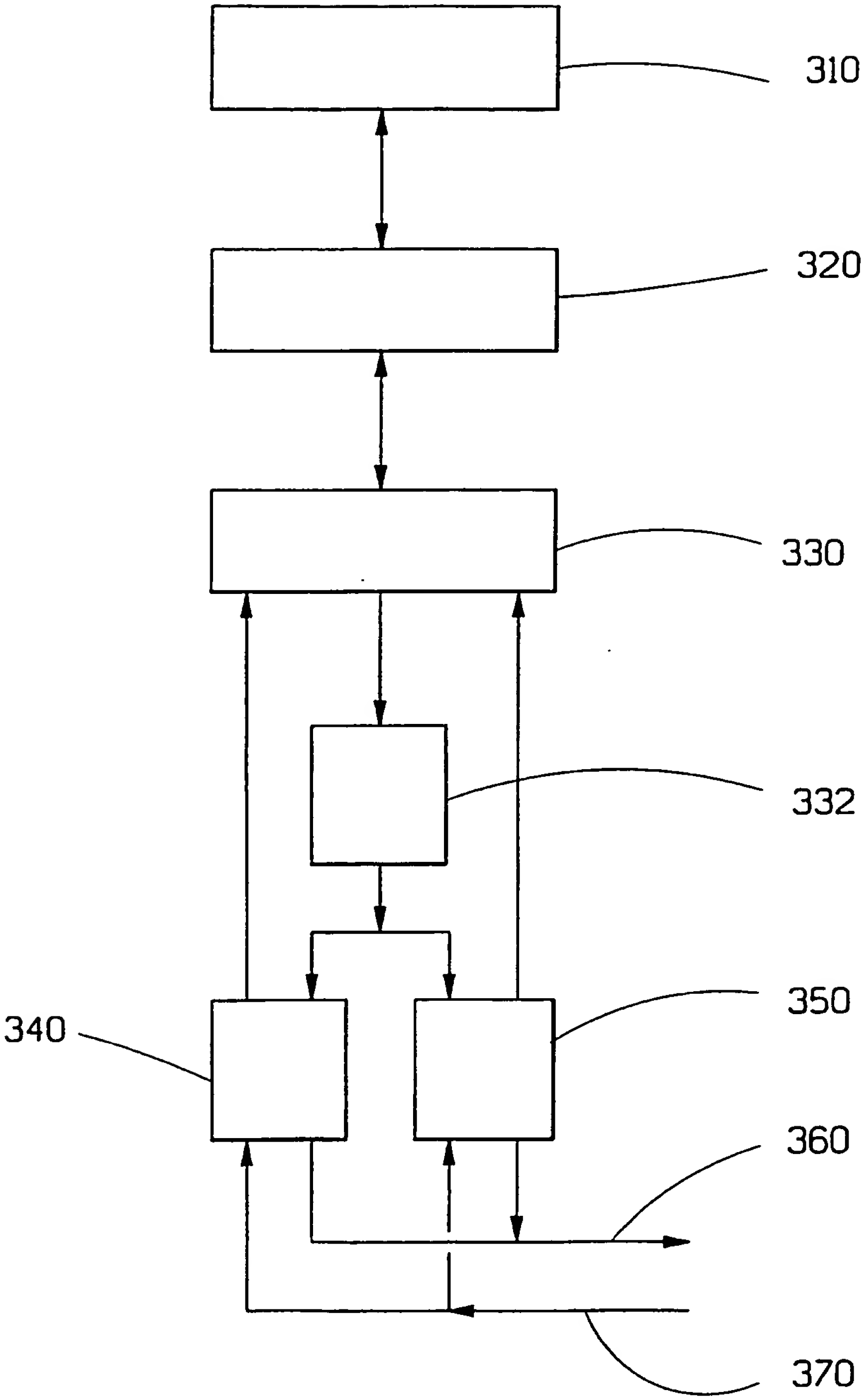


FIG.3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 00/01335

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04Q 11/04, H04L 1/00, H04L 29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04Q, H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 9837670 A1 (AT & T CORP.), 27 August 1998 (27.08.98), page 4, line 12 - page 6, line 28; page 18, line 1 - page 20, line 28, claims 1,7,8, 16-20, abstract --	1-9
A	N.SAMARAWEERA ET AL:"Integration of Internet Traffic with Digital Video Broadcasting"; UK Teletraffic Symposium (UKTS), IEEE, 1998. See section 4-5, abstract. --	1-9

☒ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" erlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 17 October 2000	Date of mailing of the international search report 18 -10- 2000
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Kristoffer Ogebjer/LR Telephone No. +46 8 782 25 00

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	"High Speed Internet Access Using Satellite-Based DVB Networks." International Network Conferens 199 pp23-28, Playmoth, UK Nihal K.G. Samaraweera and Godred Fairhurst Electronics Research Group, Department of Enigineering University of Aberdeen, Aberdeen, AB24 3UE, UK Abstract, section 3.2 --	1-9
PA	Internet Engineering Task Force INTERNET DRAFT draft-floyd-sack-00.txt Sally Floyd, Jamshid Mahdavi, Matt Mathis, Matthew Podolsky, Allyn Romanow, August 1999 An Extension to the Selective Acknoledgement (SACK) Option for TCP See the whole document. --	1-9
A	RFC2018 Network Working Group Request for Comments: 2018 Category: Standards Track. M. Mathis, J. Mahdavi, S. Floyd, LBNL, A. Romanow Sun Microsystems October 1996. TCP Selective Acknowledgment Options See the whole document. --	1-9
A	GOYAL,R. ET AL: "TCP Selective Acknowledgements and UBR Drop Policies to Improve ATM-UBR Performance over Terrestrial and Satellite Networks"; Computer Communications and Networks, 22-25 September1997. Proce International Conference on; IEEE, on 16 October 2000; see pages 17 - 25 --	1-9
A	GOYAL,R. ET AL:"Providing Rate Guarantees to TCP over the ATM GFR Service"; Local Computer Networks, 1998. LCN '98. Proceedings Conference on; 11 - 14 October 1998; IEEE, 16 October2000; See pages 390 - 398. -- --------	1-9

INTERNATIONAL-TYPE SEARCH REPORT

Search request No.

PCT/SE 00/01335

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Providing Rate Guarantees to TCP over the ATM GFR Service. Rohit Goyal, Raj Jain, Sonia Fahmy, Bobby Vandalore, Department of computer information science. The Ohio State University. 2015 Neil Avenye, DL 395 Columbus, OH 43210 0-8186-8810-6/98 $10.00 8k 1997 IEEE Section 4-5, abstract -- --------	1-9

INTERNATIONAL-TYPE SEARCH REPORT

Information on patent family members

Search request No.

PCT/SE 00/01335

WO	9837670 A1	27/08/98	US	5974028 A		26/10/99